STM32F10x Simulation with Keil uVision

Objective: Familiarisation - Keil environment.

---------------------------------------------------------------------------------------------------------------------

General Information:

* Simulation port: GPIOB(PB.0-PB.15) configured as output
* Setting the individual bit: BSRR of GPIOB
* Resetting the individual bit: BRR of GPIOB
* Output data register: ODRof GPIOB (Word)

---------------------------------------------------------------------------------------------------------------------

Lab Exercise:

1. Write a program to turn on the (PB0)
2. Write a program to turn on the (PB2)
3. Modify the program to turn off the (PB0)
4. Write a program to turn on the (PB0-PB7)
5. Write a program to turn on (PB2) using ODR
6. Write a program to turn on (PB1 and PB2) using ODR
7. Write a program to turn on (PB3) using ODR ?
8. Write a program to turn on (PB0-PB3) using ODR and turn off (PB0-PB2) ie PB3 keep it in on stage.

**Answers to the tasks:-**

#include <stm32f10x.h>

void wait (void) {

int d;

for (d = 0; d < 200; d++);

}

//Task 1

void LED1\_on (void) {

unsigned int i;

RCC->APB2ENR |= (1UL << 3);

GPIOB->CRH = 0x33333333;

GPIOB->CRL = 0x33333333;

while (1) {

i=0;

GPIOB->BSRR=i;

wait();

}

}

//Task 2

void LED3\_on (void) {

unsigned int i;

unsigned int i;

RCC->APB2ENR |= (1UL << 3);

GPIOB->CRH = 0x33333333;

GPIOB->CRL = 0x33333333;

while (1) {

i=4;

GPIOB->BSRR=i;

wait();

}

//Task 3

void LED1\_off (void) {

unsigned int i;

RCC->APB2ENR |= (1UL << 3);

GPIOB->CRH = 0x33333333;

GPIOB->CRL = 0x33333333;

while (1) {

i=0;

GPIOB->BRR=i;

wait();

}

}

//Task 4

void LED\_all (void) {

unsigned int i;

RCC->APB2ENR |= (1UL << 3);

GPIOB->CRH = 0x33333333;

GPIOB->CRL = 0x33333333;

while (1) {

for (i = 1; i < 255; i <<= 1) {

GPIOB->BSRR = i;

}

wait ();

}

}

void main (void) {

LED1\_on();

LED3\_on();

LED1\_off();

LED\_all();

}

//Task 5

unsigned int a;

#define RCC\_APB2ENR (int \*) (0x40021000+0x18)

#define GPIOB\_CRL (int \*) (0x40010C00+0x00)

#define GPIOB\_ODR (int \*) (0x40010C00+0x0C)

int main (void) {

\*RCC\_APB2ENR|=0x08;

\*GPIOB\_CRL|=0x01;

while(1) {

a++;

\*GPIOB\_ODR|=0x04;

wait();

}

}

//Task 6

unsigned int a;

#define RCC\_APB2ENR (int \*) (0x40021000+0x18)

#define GPIOB\_CRL (int \*) (0x40010C00+0x00)

#define GPIOB\_ODR (int \*) (0x40010C00+0x0C)

int main (void) {

\*RCC\_APB2ENR|=0x08;

\*GPIOB\_CRL|=0x01;

while(1) {

a++;

\*GPIOB\_ODR|=0x02;

\*GPIOB\_ODR|=0x04;

wait();

}

}

//Task 7

unsigned int a;

#define RCC\_APB2ENR (int \*) (0x40021000+0x18)

#define GPIOB\_CRL (int \*) (0x40010C00+0x00)

#define GPIOB\_ODR (int \*) (0x40010C00+0x0C)

int main(void)

{

\*RCC\_APB2ENR|=0x08;

\*GPIOB\_CRL|=0x01;

while(1)

{

unsigned int i;

a++;

\*GPIOB\_ODR|=0x08;

wait();

}

}

//Task 8

unsigned int a;

#define RCC\_APB2ENR (int \*) (0x40021000+0x18)

#define GPIOB\_CRL (int \*) (0x40010C00+0x00)

#define GPIOB\_ODR (int \*) (0x40010C00+0x0C)

int main(void)

{

\*RCC\_APB2ENR|=0x08;

\*GPIOB\_CRL|=0x01;

while(1)

{

unsigned int i;

a++;

for(i=0x02; i<=0x08; i<<=0x01) {

\*GPIOB\_ODR|=i;

}

wait();

\*GPIOB\_ODR=0x08;

wait();

}

}